

Eur. J. Clin. Chem. Clin. Biochem.
Vol. 31, 1993, pp. 311–316

© 1993 Walter de Gruyter & Co.
Berlin · New York

Self-Organizing Neural Networks as a Means of Cluster Analysis in Clinical Chemistry

By G. Reibnegger, G. Weiss and H. Wachter

Institute of Medical Chemistry and Biochemistry, University of Innsbruck, Innsbruck, Austria

(Received October 26, 1992/February 9, 1993)

Summary: Connectionist systems (often termed “neural networks”) are an alternative way to solve data processing tasks. They differ radically from conventional “von-Neumann” computing devices. Recent work on neural networks in clinical chemistry was done using supervised learning schemes, resulting in models which resemble classical discriminant analysis. The aim of the present study is to make clinical chemists familiar with basic concepts of self-organizing neural networks employing unsupervised learning schemes. Using a benchmark data set on the composition of milk from 22 different mammals, it is demonstrated that self-organizing neural networks are capable of performing tasks similar to classical cluster analysis and principal component analysis. Self-organizing neural networks could be envisaged to provide an alternative way for reducing the dimensionality of complex multivariate data sets, thus producing easily comprehensible low-dimensional “maps” of essential features.

Introduction

The basic concepts of connectionist computing schemes, often referred to in a suggestive way as “neural networks”, date back to the 1940’s (1). An initial burst of enthusiasm accompanied the invention of the “perceptron” (2), but early hopes were disappointed, and only few pioneers continued studying such models (3). The demonstration that principles of the theoretical physics of multiparticle systems are applicable to connectionist data processing schemes (4) gave new respectability to the field, and the invention of ingenious and efficient learning schemes such as the error back-propagation method (5) led to a remarkable renaissance of interest in the field.

Neural networks are an attempt to model, albeit in a very primitive and over-simplistic way, principles of data processing which are thought to give biological nervous systems and brains their superb capabilities in performing complex tasks, particularly in the wide and important field of pattern recognition. It is only natural therefore that neural network models have also been studied in medical science, particularly with

the aim of recognizing patterns underlying complex data sets and employing these patterns for diagnostic purposes (6–12).

Most if not all applications of artificial neural networks in medicine published so far have made use of a “supervised” learning scheme; i.e. a training data set is presented to a network together with a desired network output, and the network, by virtue of the learning rules applied, “learns” to produce the correct output. In conventional terms, the network is trained to perform a task resembling discriminant analysis: a given input data vector has to be transformed into an output vector representing the correct class membership of the input data vector.

However, neural networks can easily be designed for “unsupervised” learning: a series of input data vectors is again presented to the network, but no output vectors are provided. Rather, in a self-organizing process driven only by the input data and the learning scheme applied, the network adjusts its internal structure without reference to an external “teacher”. It can be shown that appropriately designed self-organizing

networks are capable of performing tasks similar to classical cluster analysis or principal component analysis; i.e., structures hidden in the complex array of input data can be extracted and visualized.

In the present paper, we aim at introducing some basic concepts of unsupervised learning by self-organizing neural networks, and we demonstrate their possible use in the realm of clinical chemistry by means of a simple example.

Materials and Methods

Data

The purpose of this work is to demonstrate the features and abilities of self-organizing neural networks. Therefore, we have deliberately chosen to use a well-studied data set (tab. 1) on the percentages of water, protein, fat and lactose in milk samples from 22 different mammals (13). These data have been used as a benchmark data set in the biostatistical software package BMDP (BMDP Statistical Software, Cork, Ireland), where they serve to illustrate cluster analysis as embodied in the program BMDP2M ("cluster analysis of cases"). Table 1 shows the data for convenience.

Network models studied

Self-organizing feature maps of the "Kohonen" type

Firstly, self-organizing feature maps or "Kohonen maps" (14) were studied. These networks are designed to map, or adaptively project, input signals of arbitrary dimensionality onto a structured set of processing units, in such a way that topological (or, more generally expressed, neighbourhood) relationships of the input patterns and of the representation patterns are kept similar.

In brief, the idea is as follows (fig. 1): suppose we have a linear array of k input neurons. In a physiological analogy, these are often referred to as "retina neurons", since they establish the contact between the network and the "environment". The number k is chosen according to the problem under consideration. For example, if we wished, as in the present study, to explore the neighbourhood relations in a data set consisting of four laboratory variables per case, we use four ($k = 4$) input neurons. Furthermore, suppose there is a two-dimensional rectangular $m \times n$ -layer of additional neurons (the "Kohonen layer"). Each of the $m \times n$ Kohonen-neurons is connected with each of the input neurons, and the connection strength ("synaptic strength") of each Kohonen neuron, j , to each input neuron, i , is represented by a scalar weight factor, $w(j,i)$. Importantly, the neurons in the Kohonen layer are characterized by their physical position in the layer, and therefore, neighbourhood relations are defined for each of these neurons.

Initially, the weights, $w(j,i)$, are randomly assigned. The input data have to be preprocessed in a way that input values with greatly different numerical values are avoided. This can easily be achieved by computing, for each variable, the mean value and standard deviation and using these to calculate "Z-scores". We have chosen, for simplicity, a linear projection of the values onto the range [0.1, 0.9]. This was done in the following way: first, the minimum (x_{\min}) and maximum value (x_{\max}) was determined. Each value x_{old} was then transformed by

$$x_{\text{new}} = (x_{\text{old}} - x_{\min}) / (x_{\max} - x_{\min}).$$

Thus, the values were linearly projected onto the interval [0, 1]. The final step consisted simply of multiplying these transformed values by 0.80 and adding 0.1.

Tab. 1. Composition of milks from different mammals (given in percent)*

Animal	Code	Water	Protein	Fat	Lactose
Horse	A	90.1	2.6	1.0	6.9
Donkey	B	90.3	1.7	1.4	6.2
Whale	C	64.8	11.1	21.2	1.6
Zebra	D	86.2	3.0	4.8	5.3
Guinea pig	E	81.9	7.4	7.2	2.7
Rat	F	72.5	9.2	12.6	3.3
Sheep	G	82.0	5.6	6.4	4.7
Reindeer	H	64.8	10.7	20.3	2.5
Mule	I	90.0	2.0	1.8	5.5
Pig	J	82.8	7.1	5.1	3.7
Camel	K	87.7	3.5	3.4	4.8
Buffalo	L	82.1	5.9	7.9	4.7
Fox	M	81.6	6.6	5.9	4.9
Rabbit	N	71.3	12.3	13.1	1.9
Llama	O	86.5	3.9	3.2	5.6
Deer	P	65.9	10.4	19.7	2.6
Hippopotamus	Q	90.4	0.6	4.5	4.4
Bison	R	86.9	4.8	1.7	5.7
Cat	S	81.6	10.2	6.3	4.4
Dog	T	76.3	9.3	9.5	3.0
Seal	U	46.4	9.7	42.0	0.0
Dolphin	V	44.9	10.6	34.9	0.9

* Data from l.c. (13)

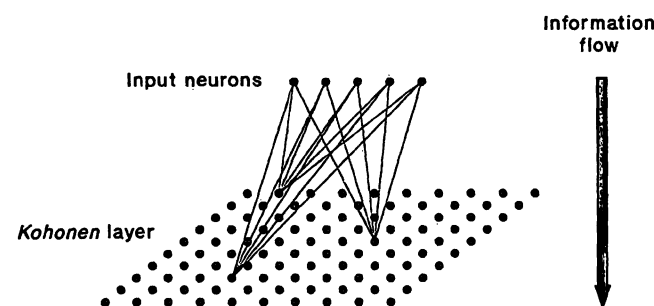


Fig. 1. Basic architecture of a Kohonen-type neural network (only few of the connections between input neurons and Kohonen-layer neurons are shown for simplicity).

If a (preprocessed) input vector is sent through the network, corresponding activity occurs in the network. The activity, a_j , of each Kohonen neuron j is given by the scalar product between its weight vector, $w(j,i)$, and the input vector, $x(i)$. In other words, each Kohonen neuron, j , computes a weighted sum

$$a_j = \sum_{i=1}^k w(j,i) \times x(i).$$

In Kohonen's simplified technique, the neuron responding maximally to a given input vector is chosen to be the neuron, the weight vector of which has the smallest Euclidean distance to the input vector. And it is this neuron which is allowed to "learn" after presentation of a given input vector: learning is achieved by changing the weights of that maximally activated neuron in a way that the Euclidean distance between its weight vector and the given input vector is further decreased. In other words, the new weight factors, $w(j,i)$, of this neuron j are given as

$$w(j,i)_{t+1} = w(j,i)_t + g(t) [x(i) - w(j,i)_t],$$

where $g(t)$ denotes the fractional increment of the correction; typically, $g(t)$ is large (approaching 1.0) for early steps of the learning process (t small), and decreases continuously, finally approaching 0.0.

In practical simulations, not only the maximally activated neuron is allowed to adjust its weight factors upon presentation of an input vector ("winner-take-all" strategy), but also the neighbouring neurons are allowed to learn, albeit to a smaller extent. The radius defining "neighbourhood" is usually set to larger values in the start phase of the learning process, and is gradually reduced; in the final phase only the maximally activated neuron is allowed to adjust its weights (to "learn") per one input vector presentation.

The final result of iterative process is that neurons in the *Kohonen* layer which are in mutual physical neighbourhood in the *Kohonen* layer, will collectively respond only and maximally to input vectors which are "neighbours" in the multi-dimensional input data space. Thus, a mapping of a possibly multi-dimensional data space onto a two-dimensional grid of neurons is achieved in a way that neighbourhood relations remain preserved to a great extent.

Importantly, the network adjusts its internal weight factors autonomously, without reference to an external teacher. The training process is driven only by the presentation of the input data and by the described simple learning rule. This is in marked contrast to the more familiar networks using the error back-propagation technique, which have been used in most applications of neural networks in medical research so far.

In order to visualize the results of the autonomous self-organization process, the input vectors were presented to the network again, after training. For each *Kohonen* neuron the input vector eliciting the strongest activity of this neuron was recorded. Moreover, the actual activity value attained by this neuron upon the optimally stimulating input vector was also calculated. This information was used for graphical representation.

Self-organization of the hidden layer in a three-layer feed-forward network employing error back-propagation

The second approach that we studied used the conventional error back-propagation technique, but this was not applied in the usual way. The details of the error back-propagation technique have been explained, for example, in a paper published (12); so we need not reiterate these issues at length.

We used the following network architecture: an input layer consisting of four input neurons (according to the number of four variables per case), two hidden neurons, and four output neurons (fig. 2). As input vectors we chose, naturally, the values of the four component variables of the milk samples. To avoid numerical problems with too large or too small arguments for the necessary exponentials, we transformed the data in just the same way as described above onto the interval [0.1, 0.9]. As output values to be used in the training phase of the network, however, we did not use, e.g., a desired class membership; in contrast, we chose that the output vectors should be equal to the input vectors. Thus, we did not introduce an external classification, although using supervised learning.

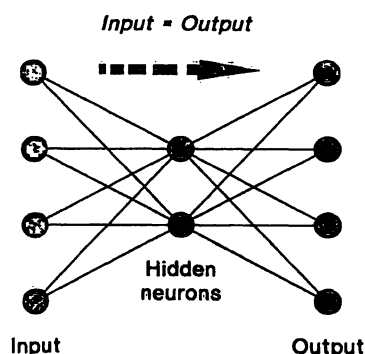


Fig. 2. Architecture for studying the behaviour of two hidden neurons in an error back-propagation network.

Obviously, the network was thus forced to learn to perform a somewhat surprising task: a given input vector has to be passed through the "bottleneck" of two hidden neurons, and the resulting output has to be a vector equal to the input vector. What is the idea underlying this exercise?

We were in fact interested in the behaviour of the two hidden neurons. *Baldi & Hornik* (15) demonstrated that a network consisting of n input units, p hidden units, and n output units, and having linear instead of non-linear activity propagation characteristics, has two very important features: there is only one global minimum of the error function, and if the minimum error state is achieved, the p weight vectors of the p hidden units represent the p first principal components with maximum eigenvalues of the covariance matrix. The more complicated case of non-linear network characteristics is not yet fully resolved theoretically, but it is thought that the hidden units are performing an action similar to principal component analysis, even in the case of the usually employed sigmoidal output functions (16).

After training the network using the error back-propagation as described, the input vectors were again presented to the trained network, and the activities attained by the two hidden neurons upon each input vector were plotted. Thus, the central layer of the two hidden neurons was used to provide the x - and y -coordinates of cases for two-dimensional display.

Comparison with conventional statistical techniques

The data were analysed for comparison using program BMDP2M of the BMDP software package. This program performs a hierarchical cluster analysis of cases, based on similarities of the variables associated with each case. The *Euclidean* distance using standardized data was chosen as the measure of similarity. Initially each case is considered as a separate cluster. In a stepwise process, cases and/or clusters are joined until all cases are combined into one single cluster. The algorithm uses the distance between centroid clusters or the k th nearest neighbor density estimator as a criterion for amalgamating clusters.

Results

I. Cluster analysis of cases

Figure 3 shows the results using conventional cluster analysis of cases. This plot demonstrates that the animals tested comprised two major clusters, each of which could be split into two smaller clusters, containing 8, 8, 2, and 4 animals which share a similar composition of milk. A fine structure further splitting these clusters into smaller groups is also visible.

II. *Kohonen* network

A *Kohonen* network consisting of 24×24 neurons, each neuron connected to four input neurons, was exposed to the milk data and trained on them. Because of the random initialization of the weight factors, subsequent rounds of training the network normally produce slightly different results, but nevertheless, essentially the same type of clustering normally appears on the trained nets. A typical result is shown in figure 4: to visualize the resulting net structure, a letter is shown for each neuron in the net, representing

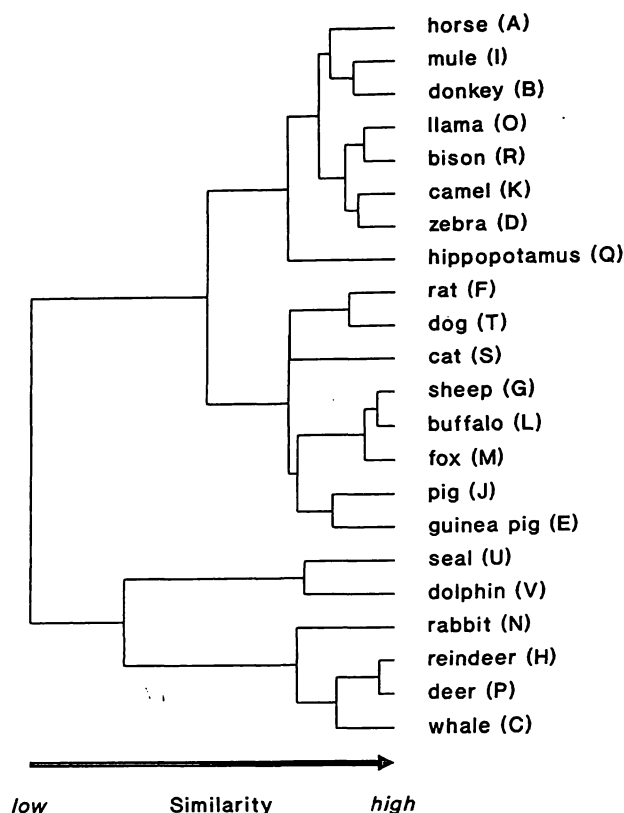


Fig. 3. Cluster analysis of cases (program BMDP2M) for the milk data shown in table 1. For letter symbols see table 1.

the animal which most strongly activates this neuron, provided this maximum activation exceeds the value of 0.99 (activity values may be in the range between 0.0 and 1.0 because a sigmoid transformation is applied to the weighted sum that each neuron computes).

It is easily seen in this drawing that neurons responding optimally to certain input vectors cluster together. On closer inspection, one detects a striking similarity of the clusters shown in figure 3, and the clusters of neurons, identified by their preferred input vectors. Indeed, even the fine structure of figure 3 is reproduced by the network; compare, for example, the results describing the relationship between the milks of reindeer (H), deer (P), whale (C) and rabbit (N), or llama (O), bison (R), camel (K) and zebra (D).

III. Hidden layer in an error back-propagation network

As described in the Methods section, a conventional 4-2-4 error back-propagation network was trained on the input data with the unusual constraint that the output vectors equalled the input vectors. A constant learning rate of 0.30 and a constant momentum factor of 0.50 (5) were applied, and training was continued until the average error function (5) dropped below

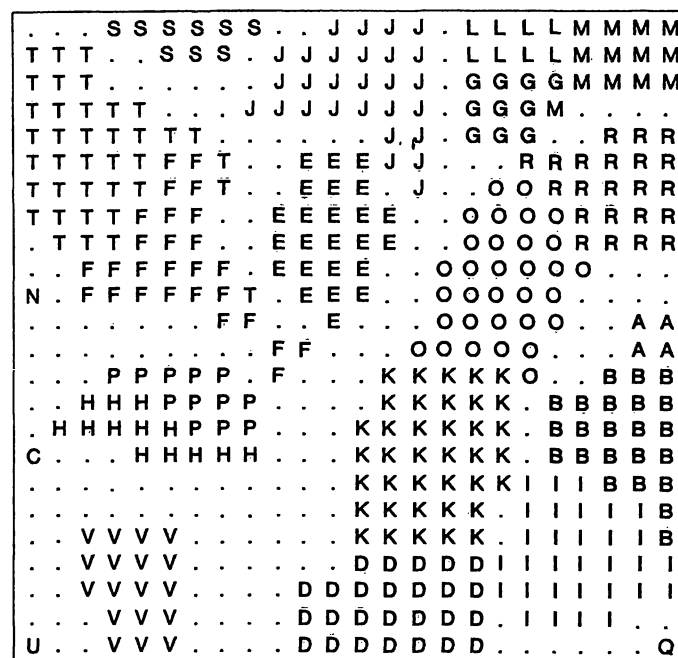


Fig. 4. Structure of the Kohonen network after self-organization. For each neuron in the Kohonen layer, the animal evoking the strongest response was recorded. If the activation attained exceeded 0.99, a letter symbol for the respective animal was drawn on the corresponding physical position of the network. For letter symbols see table 1.

0.10 (approximately 1000 training cycles were required to achieve this goal). Each input vector was then presented to the trained network, and the output values of both hidden neurons were taken as x- and y-coordinates for the scatter plot shown in figure 5.

As indicated by the rectangles, the four main clusters described above were also detected clearly by this procedure.

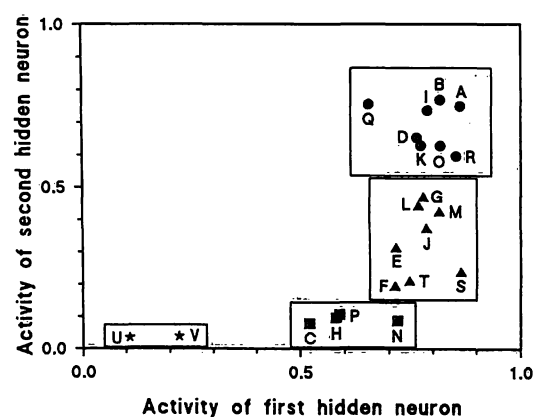


Fig. 5. Activation attained by the two hidden neurons in an error back-propagation network trained on the animal's milk composition data. For letter symbols see table 1.

Discussion

The techniques dealt with in this paper are essentially display techniques: each of them can serve to display a readily understood synopsis of large and potentially complicated data matrices. This task may be particularly important in early stages of statistical analysis of such data sets. For the realm of clinical chemistry, the general usefulness of clustering techniques has been reviewed recently (17).

We could demonstrate that unsupervised learning techniques using connectionist computing schemes can be as useful in the field of cluster analysis and related techniques of unsupervised pattern recognition, as are their supervised counterparts in the domain of discriminant analysis and classification.

Kohonen networks are interesting types of artificial neural networks. Their development has been strongly stimulated by observations on real biological neural networks (14). It has to be noted that several useful extensions have been suggested for *Kohonen* type neural networks. For example, the so-called "counterpropagation" network shows, in addition to the input neurons and the *Kohonen* layer, a layer of output neurons, which are designed in a manner similar to output neurons of error back-propagation networks and which modify their weights, in a supervised manner, after the *Kohonen* layer has organized itself by unsupervised learning. *Hecht-Nielsen* has discussed the properties of such hybrid networks and has given useful examples for applications (18).

The technique of using the output values of the hidden neurons of an error back-propagation network for

low-dimensional display of multivariate data matrices has been described, albeit in a slightly more complicated version, in the domain of quantitative structure-activity relationships in medicinal chemistry (19). These authors have also compared the technique with principal component analysis and non-linear mapping techniques. They found the results of the neural network technique to be comparable or even superior to those of the more conventional techniques. We have recently further clarified some issues related to this novel technique (20).

It might well be that for very complex data structures a two-dimensional display method is insufficient, because of remaining overlap between clusters. The neural network technique using the output of hidden neurons for display could of course easily be modified by using, e.g., three instead of two hidden neurons. Their output values could then be taken as x-, y- and z-components for a three-dimensional plot, thus possibly resolving the overlap between clusters.

Neural network approaches for processing of complex data matrices seem to offer interesting avenues. Besides error back-propagation networks and the network types mentioned in this work, there are in fact numerous other possible network architectures which might offer valuable tools for variable tasks. Further research probing these tools in connection with real data sets, and thorough comparison of the results with those of classical statistical techniques, are urgently required to allow a fair judgement of their value, and thence exploitation of them, in clinical chemistry.

References

- McCulloch, W. S. & Pitts, W. H. (1943) A Logical Calculus of the Ideas Imminent in Neural Nets. *Bull. Math. Biophys.* 5, 115–133.
- Rosenblatt, F. (1962) *Principles of Neurodynamics*. Spartan, New York.
- Cooper, L. N. (1973) A Possible Organization of Animal Memory and Learning, In: *Proceedings of the Nobel Symposium on Collective Properties of Physical Systems* (Lundquist, B. & Lundquist, S., eds.) pp. 252–264, Academic Press, New York.
- Hopfield, J. J. (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. USA* 79, 2554–2558.
- Rumelhard, D. E., Hinton, G. E. & Williams, R. J. (1986) Learning Internal Representations by Error Propagation, In: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition* (Rumelhard, D. E. & McClelland, J. L., eds.) pp. 318–364, MIT Press, Cambridge, Massachusetts.
- Boone, J. M., Gross, G. W. & Greco-Hunt, V. (1990) Neural Networks in Radiologic Diagnosis. I. Introduction and Illustration. *Invest. Radiol.* 25, 1012–1016.
- Bounds, D. G., Lloyd, P. J. & Mathew, B. G. (1990) A Comparison of Neural Network and Other Pattern Recognition Approaches to the Diagnosis of Low Back Disorders. *Neural Networks* 3, 583–591.
- Gross, G. W., Boone, J. M. & Greco-Hunt, V. (1990) Neural Networks in Radiologic Diagnosis. II. Interpretation of Neonatal Chest Radiographs. *Invest. Radiol.* 25, 1017–1023.
- Mulsant, B. H. (1990) A Neural Network as an Approach to Clinical Diagnosis. *MD Computing* 7, 25–36.
- Baxt, W. G. (1991) Use of an Artificial Neural Network for the Diagnosis of Myocardial Infarction. *Ann. Intern. Med.* 115, 843–848.
- Reibnegger, G., Werner-Felmayer, G., Judmaier, G. & Wachter, H. (1991) Neural Networks as a Tool for Utilizing Laboratory Information: Comparison with Linear Discriminant Analysis and with Classification and Regression Trees. *Proc. Natl. Acad. Sci. USA* 88, 11426–11430.
- Astion, M. L. & Wilding, P. (1992) Application of Neural Networks to the Interpretation of Laboratory Data in Cancer Diagnosis. *Clin. Chem.* 38, 34–38.
- Hartigan, J. A. (1975) *Clustering Algorithms*, p. 304, Wiley, New York.

14. Kohonen, T. (1984) *Self-Organization and Associative Memory*, Springer, Berlin.
15. Baldi, P. & Hornik, K. (1989) Neural Networks and Principal Component Analysis. *Neural Networks* 2, 53–58.
16. Brause, R. (1991) *Neuronale Netze*, pp. 114–117, Teubner, Stuttgart.
17. Vogt, W. & Nagel, D. (1992) Cluster Analysis in Diagnosis. *Clin. Chem.* 38, 182–198.
18. Hecht-Nielsen, R. (1988) Applications of Counterpropagation Networks. *Neural Networks* 1, 131–139.
19. Livingstone, D. J., Hesketh, G. & Clayworth, D. (1991) Novel Method for the Display of Multivariate Data Using Neural Networks. *J. Mol. Graphics* 9, 115–118.
20. Reibnegger, G., Werner-Felmayer, G. & Wachter, H. (1993) A Note on the Low-Dimensional Display of Multivariate Data Using Neural Networks. *J. Mol. Graphics* 11, in press (March 1993).

Univ.-Doz. Dr. Gilbert Reibnegger
Institut für Medizinische Chemie und Biochemie
Universität Innsbruck
Fritz-Pregl-Straße 3
A-6020 Innsbruck
Republik Österreich